

RECEIVED
CENTRAL FAX CENTER

NOV 01 2006

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the Application of) Before the
Appellants: Alok Srivastava et al.) Board of Patent Appeals
Serial No. 09/871,440) and Interferences
Filed: May 31, 2001) Examiner
Title: XML Aware Logical Caching System) Joseph E. Avellino
Art Unit 2143)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

FAX TRANSMISSION COVER LETTER

- 20 SHEETS TOTAL -

The attached APPELLANTS' BRIEF and Claims Appendix (18 pages) and Credit Card form in the amount of \$500 for the fee for filing the Brief are being transmitted via facsimile to the central facsimile number of United States Patent and Trademark Office, (571) 273-8300, on November 1, 2006.

Respectfully submitted,



Dated: November 1, 2006

Charles G. Call, Reg. 20,406
USPTO Customer No. 021253
68 Horse Pond Road
West Yarmouth, MA 02673-2516
Fax: (508) 629-6540

RECEIVED
CENTRAL FAX CENTER

NOV 01 2006

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the Application of) Before the
Appellants: Alok Srivastava et al.) Board of Patent Appeals
Serial No. 09/871,440) and Interferences
Filed: May 31, 2001)
Title: XML Aware Logical Caching System) Examiner
Joseph E. Avellino
Art Unit 2142

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

APPELLANTS' BRIEF

Real party in interest

The real party in interest is Oracle International Corporation, 500 Oracle Parkway,
Redwood Shores, CA 94065 USA.

Related appeals and interferences

None.

Status of claims

Claims 1-18 are pending and have been finally rejected and are appealed.

Status of amendments

No amendments were filed subsequent to final rejection.

Summary of claimed subject matter

The particular sections of the specification and drawings referenced below are illustrative and representative, but do not identify all of the descriptive material that corresponds to any element of the appealed claims.

The present invention set forth in independent claims 1, 6, 8, 10, 15 and 17 takes the form of methods and apparatus for responding to an incoming request message from a sender [Fig. 1 at 101 and 103; page 3, lines 2-7; page 4, line 23 through page 6, line 9]. The incoming request message is converted into incoming canonical request message expressed in a predetermined standard form [Fig. 1 at 105 page 3, lines 7-10 and 20-26; page 7, line 1 through page 8, line 7]. The incoming canonical request message is then compared with previously received and stored canonical request messages to determine if said incoming request message is logically equivalent to one of the previously received and stored canonical request messages [Fig. 1 at 107, 109 and 133; page 3, lines 9-19; page 8, lines 8-26]. If a match is found between the incoming canonical request message and a given previously stored canonical request message, a stored response previously transmitted in response to the given previously stored canonical message is accessed and returned to the sender [Fig. 1 at 133; page 3, lines 10-11].

Independent claims 6, 8, 15 and 17 further specify that the incoming request message is expressed in the Extended Markup Language (XML) [page 7, line 1 through page 8, line 7] and independent claims 8 and 17 further specify that an incoming message is an HTTP message [page 4, line 23 through page 6, line 9].

Grounds of rejection to be reviewed on appeal

Ground 1 (Claims 1, 3-5, 10, and 12-14): Claims 1, 3-5, 10, and 12-14 were finally rejected under 35 U.S.C. §103(a) as being directed to subject matter deemed to be obvious in view of the combined teachings of U.S. Patent 6,292,880 issued to Mattis et al. (hereinafter "Mattis") and U.S. Patent 6,594,700 issued to Graham et al. (hereinafter "Graham").

Ground 2 (Claims 2, 6-9, 11, and 15-18): Claims 2, 6-9, 11, and 15-18 were finally rejected under 35 U.S.C. §103(a) as being unpatentable over Mattis in view of Graham, and further in view of U.S. Patent Application Publication No. 2002/0099734 filed by Schroeder et al. (hereinafter "Schroeder").

Argument

The two grounds of rejection presented for review are discussed individually below.

Ground 1: The obviousness rejections based on Mattis in view of Graham

The appealed claims describe an improved caching mechanism which delivers a cached response to any incoming data request that is logically equivalent to a prior request, even though the prior request may have different character content. As claimed, appellants' invention first converts each incoming request message into a standard canonical form and then compares that canonical incoming message with previously received and stored canonical requests. If a match is found, a cached response data that was previously transmitted in response to the previously received matching request is returned.

Appellants' invention avoids the need to perform transaction processing to respond to a received request that is logically the same as but not identical to a prior request. Appellants' claimed invention allows a cached response to be returned whenever an incoming request is logically equivalent to a cached canonical request, even though its character content may differ from the original request. This in turn enables the system to immediately return cached responses without any additional processing, avoiding the need to access and package response data into a return message and avoiding the need to cache a further return message that duplicates a message that has already been cached.

Appellants' invention is an improvement over caching systems of the type taught by Mattis which do not convert incoming messages into canonical form before processing. Mattis describes an advanced caching mechanism which, like appellants' invention, is used to cache HTTP requests and responses of the kind received and returned by a web server. The "Background" section at columns 1-5 of Mattis contains an extensive discussion of the conventional web server caching systems. Mattis processes requests which identify desired objects by name (i.e., a file system name, a network address or a URL as explained at col. 10, lines 3-4) by first converting the object name into a name key that is used to locate an object key index to the desired object content (see col. 8, lines 4-36). This keyed access mechanism permits high speed comparisons to be made.

But, as acknowledged by the Examiner, Mattis does not disclose or suggest converting incoming messages into canonical form for processing. When comparing a received request with a different but logically equivalent cached request, the Mattis caching system will not return a cached response but will instead trigger a new data processing and output message formatting process, and will perform duplicative caching of the second request and the new response.

The Examiner contends that it would have been obvious to modify the Mattis caching system to incorporate the teachings of Graham.

Graham describes a Universal Service Broker Interchange Mechanism (USBIM) which permits client processes that use many different protocols to obtain service advertisements from service providers that also use many different protocols. Each service provider employs a service protocol adapter servlet to translate its service advertisement a canonical form (preferably an XML document which conforms to a predetermined Document Type Definition) before it is posted into an advertisement registry. In this way, all of the service advertisements from all of the service providers, regardless of the protocol used by the individual service providers, are stored in a standard format in the advertisement registry. When a client issues a request for an advertisement using its native protocol, that request is converted into a standard canonical form that is compatible with the form in which the advertisements are stored.

But it would not have been obvious to one of ordinary skill in the art to modify the Mattis caching system in view of Graham to place incoming requests in canonical form as suggested by the Examiner. A person of ordinary skill in art would have no reason to attempt such a modification.

It is often said that "Necessity is the mother of invention."¹ But before one skilled in the art can provide a solution to a problem, he or she must be aware of the problem, or be made aware of the problem and solution from some source. But Mattis nowhere indicates an awareness of the problem that a different request that is logically equivalent to a prior request will not produce a cached response.

Graham likewise does not suggest that problem or its solution. While Graham converts incoming request messages into canonical form, it does so for a completely different reason and

¹ The saying appears in the dialogue *Republic*, by the ancient Greek philosopher Plato. *The New Dictionary of Cultural Literacy, Third Edition*, 2002, Houghton Mifflin.

in a way that is unlike the mechanism claimed by appellants. Graham converts requests using different protocols into standard canonical form so that service advertisements can be retrieved. Graham's canonicalization is not performed to cache data, the canonical requests are not stored as claimed, nor are canonical requests compared with prior requests.

In short, if one skilled in the art considered Mattis and Graham, singly or in combination, they would find no disclosure whatsoever of either the problem solved by appellants' invention or the solution claimed. The person of ordinary skill would find not find any suggestion in either reference that prior art caching systems fail to properly handle requests that are logically equivalent but have different character content. Moreover, the system taught by Graham for permitting service advertisements to be retrieved when the service providers and clients use different protocols has nothing to do with caching and does not deal with either the problem or the solution.

As stated §2142 of the Manual of Patent Examining Procedure:

"The legal concept of *prima facie* obviousness is a procedural tool of examination which applies broadly to all arts.

The examiner bears the initial burden of factually supporting any *prima facie* conclusion of obviousness. If the examiner does not produce a *prima facie* case, the applicant is under no obligation to submit evidence of nonobviousness.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations."
(emphasis added)

It is submitted that the Examiner's obviousness rejection fails to satisfy any of these three criteria.

There is no suggestion or motivation in either Mattis or Graham that would motivate one of ordinary skill to combine the protocol handling capabilities of Graham with Mattis' caching system. Mattis teaches a system for caching HTTP web server requests and has no need to handle requests from multiple protocols. Neither Mattis nor Graham recognize the need for a better way to cache logically equivalent requests which have different content and neither proposes the combination of functions (canonicalizing incoming requests, comparing them with previously stored canonicalized requests, and caching the incoming canonicalized request if no match is found) set forth in the claims for solving this problem.

The Mattis and Graham references plainly provide no reasonable "expectation of success" in solving a problem that neither acknowledges or attempts to deal with, and neither discloses a mechanism that would solve the problem even if it had been recognized.

In the "Response to Arguments" section of the final rejection, at page 6, the Examiner furthermore suggests that Mattis and Graham essentially provide the same service in the sense that "*both receive a request from a client, convert the request into a format with which the system can utilize (in Mattis, the name is hashed to a value; in Graham the request is formatted into an XML canonical representation), they look into a registry to find a matching entity, and then an object is returned to the client when a match is found (Mattis returns the cached object, Graham brokers a communication between the client and the service provider).*"

But the Examiner has ignored the words of the claims which require that incoming canonicalized requests be stored so that they can be compared with later arriving canonicalized requests. Graham does not store compare incoming advertising requests with previously stored requests. Graham searches the advertisement registry to find an advertisement that satisfies the incoming request; that is, it performs a search for data, not a search for a matching, previously stored canonicalized request as claimed. It is submitted that, in all the ways that count, the Mattis cache and the Graham service broker do not "essentially provide the same service" as suggested by the Examiner. Indeed, the services are quite different and one of ordinary skill would have no reason to believe that Graham offered a solution to a caching problem whose existence neither reference discloses.

In the advisory action mailed May 4, 2006, the Examiner stated:

*Applicant's arguments presented April 9, 2006 have been fully considered but are not persuasive. In the remarks, Applicant argues, that (1) Graham does not determine if an incoming request matches a previously received stored request, rather compares requests with stored data, and therefore there is no motivation to combine Mattis and Graham, (2) Mattis and Graham provide different services and nothing but hindsight can come up with the subject matter to motivate one of ordinary skill in the art to combine the art ... * * **

As to point (1) one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See In re Keller, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); In re Merck & Co., 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). In this case, it is the combination of Graham and Mattis which provides the claimed invention. Furthermore once the request is received by the claimed invention, it then becomes "stored data", since it is stored by the caching system. By this rationale, the rejection is maintained.

But appellants do not "attack the references individually" by pointing out that neither reference discloses either the problem or the solution claimed by applicants. Whether taken singly or in combination, Mattis and Graham simply do not disclose the claimed invention, do not disclose methods or structures which even if combined would satisfy the express limitations of the claims, and do not even recognize the problem which would motivate one skilled in the art to seek a solution.

The Examiner's statement that "Furthermore once the request is received by claimed invention, it becomes "stored data", since it is stored by the caching system" is not understood. If the Examiner means that Graham's requests are "stored data" that is undoubtedly the case (at least temporarily); but Graham does not compare incoming requests with stored requests as claimed, but instead compares incoming requests with stored service advertisements.

The Examiner goes on to argue:

As to point (2) Mattis and Graham both provide a request-response service. Both systems receive requests, transform the request into a format which the system can utilize, and then look into a registry to find a requested entity, and then return an implementation of that entity. Furthermore it must be recognized that any judgment on obviousness is in a

sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See In re McLaughlin, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971). One of ordinary skill in the art at the time the invention was made was knowledgeable in the fields of protocol conversion as well as caching and therefore would have had the skill to combine the two teachings. By this rationale, the rejection is maintained.

The Examiner's comparison of Mattis and Graham is carefully worded to suggest that the two systems do the same thing. But in fact the systems are very different in ways that the claims make important. Mattis describes a cache which compares incoming requests with prior requests and, if a match is found, returns the response to the prior request to eliminate the need to again create and format a response. But Mattis does not handle the caching of logically equivalent but different requests as claimed by appellants. Graham does not compare requests with requests at all, and canonicalizes requests in order to perform a task (looking for service advertisements that satisfy the request) that Mattis does not need to perform. One skilled in the art would have no reason to believe that Graham's solution to a problem Mattis does not need to solve would be applicable to Mattis' caching system. It is not enough to state that both systems handle requests.

The fact is, there is nothing to support the Examiner's obviousness rejection but hindsight reconstruction. Neither reference discloses appellants' invention or the need for the invention. To find that, one must read appellants' specification.

Finally, the Examiner contends that "One of ordinary skill in the art at the time the invention was made was knowledgeable in the field of protocol conversion as well as caching and therefore would have had the skill to combine the two teachings." In response, it is submitted that the Examiner has failed to find any suggestion in either reference that would motivate such a person skilled in both protocol conversion and caching to combine the two teachings to yield appellants invention as claimed. To simply assert that it would have been obvious, as the Examiner does, without identifying any teaching or suggestion in either reference to support that assertion, is inadequate to establish a *prima facia* case of obviousness as a matter of law.

The rejection of claims 1, 3-5, 10, and 12-14 as being directed to subject matter deemed to be obvious in view of Mattis and Graham should be reversed.

Ground 2: The obviousness rejection in view of Mattis, Graham and Schroeder

Claims 2, 6-9, 11, and 15-18 were finally rejected under 35 U.S.C. §103(a) as being unpatentable over Mattis in view of Graham, and further in view of U.S. Patent Application Publication No. 2002/0099734 filed by Schroeder et al. (hereinafter "Schroeder").

Applicants' claims 2, 6-9, 11 and 15-18 further specify that all or part of the request messages which are translated into canonical form are expressed in XML. As pointed out in applicants' specification, data requests expressed in XML may be logically identical but have different content; for example, logically identical XML request messages may have different line ending characters or include different whitespace characters which change the form but not the logical meaning of the request.

Applicants' claimed technique of converting incoming XML requests into canonical form for storage and comparison permits logically identical requests to be identified even though they don't have identical content as received. As explained at pages 7-8 of appellants' specification, 14 different conversion steps are performed in order to canonicalize an incoming XML document. Nothing in any of the cited references discloses or suggests canonicalizing XML request messages. As the Examiner concedes in Section 8 of the final rejection, neither Mattis nor Graham discloses that a portion of the incoming request message be expressed in XML language or that it should be translated into a standard canonical XML form. Neither the caching system described by Mattis nor the protocol broker taught by Graham deal with the special problems associated with caching XML requests.

The Examiner notes that "*Schroeder discloses an incoming data object in XML language that is translated into a standard canonical XML form.*" But the cited passage of Schroeder at paragraphs [0048] and [0049] does not deal with caching, but instead with "normalizing" received XML data objects (not requests) by passing them through standard XSL transforms. There is no suggestion in the cited passage of Schroeder that these "data objects" are requests or that, once "normalized" that they are stored or compared with previously stored prior requests. In short, there is nothing in the cited passage of Schroeder that suggests that incoming XML requests should be placed in canonical form so that they can be compared with previously stored

canonical requests to identify prior requests that are logically identical to the incoming request as claimed. Nothing in Schroeder suggests that XML requests may be logically equivalent even though they have differing character content, and nothing in Schroeder suggests that this characteristic can lead to caching inefficiencies, and nothing in Schroeder suggests a solution to this problem. Like Mattis and Graham, Schroeder contains no hint of the claimed invention.

In the advisory action, the Examiner stated that a "*data object*" can be construed as any *data which is received by a computing entity*. Therefore one of ordinary skill in the art would understand that the received data request can be treated as any received data object, with all the modifications available. By this rationale, the rejection is maintained." The Examiner's attempted justification of Schroeder's failure to teach canonicalizing XML requests on the basis that "requests can be treated as data objects" falls far short of the needed teaching that XML requests are the things canonicalized, stored and then compared with earlier canonicalized requests in order to properly handle XML requests which are logically equivalent but different in form. Schroeder teaches normalizing data (e.g. purchase order files) for further processing. Schroeder does not disclose the specific subject matter claimed, or the reasons why that subject matter exists in the first place.

It is accordingly submitted that the final rejection of claims 2, 6-9, 11, and 15-18 as being obvious in view of Mattis, Graham and Schroeder should be reversed for the reasons given above, as well as the reasons presented in connection with Ground 1.

Conclusion

The final rejection of claims 1-18 should be reversed.

Claims appendix

An appendix containing a copy of claims 1-18 involved in the appeal is attached.

Evidence appendix

No other evidence was submitted or is being relied upon by appellants.

Related proceedings appendix

There are no related proceedings.

Respectfully submitted,



Dated: November 1, 2006

Charles G. Call, Reg. 20,406

Certificate of Transmission under 37 CFR 1.8

I hereby certify that this *Appellants' Brief* is being transmitted by facsimile to the central facsimile number of the U.S. Patent and Trademark Office, (571) 273-8300, on November 1, 2006



Dated: November 1, 2006

Signature

Charles G. Call, Reg. No. 20,406
USPTO Customer No. 021253
68 Horse Pond Road
West Yarmouth, MA 02673
Ph. (508) 778-2630 Fax (508) 629-6540
call@patentssoft.com

CLAIMS APPENDIX

1. The method of responding to an incoming request message from a sender which comprises, in combination, the steps of:

converting said incoming request message into an incoming canonical request message expressed in a predetermined standard form,

comparing said incoming canonical request message with previously received and stored canonical request messages to determine if said incoming request message is logically equivalent to one of said previously received and stored canonical request message, and

if a match is found between said incoming canonical request message and a given previously stored canonical request message, accessing a stored response previously transmitted in response to said given previously stored canonical message, and returning said stored response to said sender.

2. The method of responding to an incoming request message as set forth in claim 1 wherein at least a portion of said incoming request message is expressed in the Extensible Markup Language and wherein said step of converting translates said portion into standard canonical XML form.

3. The method of responding to an incoming request message as set forth in claim 1 wherein said step of comparing comprises the substeps of:

generating an access key value based on the content of said incoming canonical request message;

accessing selected ones of said previously received and stored canonical request messages which are specified by said access key value, and

comparing said incoming canonical request message with said selected ones of said previously received and stored canonical request messages.

4. The method of responding to an incoming request message as set forth in claim 3 wherein, when no match is found between said incoming canonical request message and a

previously stored canonical request message, performing the step of storing said incoming canonical request message in a first storage location specified by said access key.

5. The method of responding to an incoming request message as set forth in claim 4 wherein, when no match is found between said incoming canonical request message and a previously stored canonical request message, performing the further steps of:

generating a new response message containing data specified by said incoming request message,

transmitting said new response message to said sender, and

storing said new response message at a second location associated with said first location.

6. The method of responding to an incoming request message expressed in the Extended Markup Language which comprises, in combination, the steps of:

receiving said incoming request message via the Internet from a remote sender,

converting said incoming request message into an incoming canonical request message expressed in an established standard format,

comparing said incoming canonical request message with previously received and stored canonical request messages to determine if said incoming request message is logically equivalent to one of said previously received and stored canonical request messages,

if a match is found between said incoming canonical request message and a given previously stored canonical request message, accessing a stored response previously transmitted in response to said given previously stored canonical message, and returning said stored response to said sender, and

if a match is not found between said incoming canonical request message and a given previously stored canonical request message, performing the steps of:

generating a new response message containing data specified by said incoming request message,

transmitting said new response message to said sender, and

storing said incoming canonical request message and said new response message at associated storage locations.

7. The method of responding to an incoming request message as set forth in claim 6 wherein said step of comparing comprises the substeps of:

generating an access key value based on the content of said incoming canonical request message;

accessing selected ones of said previously received and stored canonical request messages which are specified by said access key value, and

comparing said incoming canonical request message with said selected ones of said previously receive and stored canonical request messages.

8. The method of caching XML request messages and the responses thereto transmitted via the Internet which comprises, in combination, the steps of:

receiving an inbound HTTP message containing a request expressed in Extended Markup Language from a sender,

translating said request into an inbound canonical request expressed into an inbound canonical request expressed in a predetermined standard canonical format, and

comparing said inbound canonical request with previously stored canonical requests to determine if said incoming request message is logically equivalent to one of said previously stored canonical requests, and, if a match is found with a particular one of said stored canonical requests, returning to said sender a stored copy of a response message previously transmitted in response to said particular one of said stored canonical requests.

9. The method of responding to an incoming request message as set forth in claim 8 wherein said step of comparing comprises the substeps of:

generating an access key value based on the content of said inbound canonical request message;

accessing selected ones of said previously received and stored canonical request messages which are specified by said access key value, and

comparing said incoming canonical request message with said selected ones of said previously receive and stored canonical request messages.

10. Apparatus for responding to an incoming request message which comprises, in combination,

means for receiving said request message from a remote sender via a data communications link,

a translator for converting said incoming request message into an incoming canonical request message expressed in a predetermined standard form,

a request cache memory for storing received canonical request messages,

a comparator for matching said incoming canonical request message with previously received canonical request messages in said request cache memory to determine if said incoming request message is logically equivalent to a previously stored one of said canonical request messages,

a response cache memory,

means coupled to said comparator and responsive to a match between said incoming canonical request message and a given previously stored canonical request message for identifying a previously transmitted response to said given previously stored canonical message, and

transmission means for sending said previously transmitted response to said remote sender via said communications link.

11. The apparatus set forth in claim 10 wherein at least a portion of said incoming request message is expressed in the Extensible Markup Language and wherein translator converts said portion into standard canonical XML form .

12. The apparatus set forth in claim 10 wherein said comparator comprises:

means for generating an access key value based on the content of said incoming canonical request message;

means for retrieving selected ones of said previously received and stored canonical request messages from locations in said request cache memory which are specified by said access key value, and

means for comparing said incoming canonical request message with said selected ones of said previously received and stored canonical request messages.

13. The apparatus set forth in claim 12 further including means responsive to the condition occurring when no match is found between said incoming canonical request message and a previously stored canonical request message for storing said incoming canonical request message at a location in said request cache memory specified by said access key.

14. The apparatus set forth in claim 13 further wherein said means responsive to the condition when no match is found between said incoming canonical request message and a previously stored canonical request message further includes:

means for generating a new response message containing data specified by said incoming request message,

means for transmitting said new response message to said sender, and

means for storing said new response message in said response cache memory.

15. Apparatus for responding to an incoming request message expressed in the Extended Markup Language which comprises, in combination:

an Internet connection for receiving said incoming request message via the Internet from a remote sender,

a translator for converting said incoming request message into an incoming canonical request message expressed in an established standard format,

a cache memory for storing previously received and converted canonical request messages and corresponding previously transmitted responses to said previously received request messages,

a comparator for comparing said incoming canonical request message with said previously received and stored canonical request messages in said cache memory to determine if said incoming request message is logically equivalent to one of said previously received and stored canonical request messages,

means coupled to said comparator and responsive to a detected match between said incoming canonical request message and a given previously stored canonical request message for identifying that given previously transmitted response in said cache memory that was transmitted

in response to said given previously stored canonical request, and for transmitting said given response to said remote sender via said Internet connection.

16. The apparatus set forth in claim 15 wherein said comparator comprises:
 - means for generating an access key value based on the content of said incoming canonical request message;
 - means for retrieving selected ones of said previously received and stored canonical request messages from locations in said cache memory which are specified by said access key value, and
 - means for comparing said incoming canonical request message with said selected ones of said previously received and stored canonical request messages.

17. In combination with a Web database server, a cache memory system for storing XML request messages and the responses thereto, said cache memory system comprising, in combination,

an Internet connection for receiving HTTP request messages from and returning HTTP response messages to a remote client,

an inbound message port for receiving HTTP request messages at least some of which contain a request payload expressed in Extended Markup Language,

a translator for converting each request payload into an inbound canonical request which conforms to a predetermined standard canonical format,

a memory for storing previously received inbound canonical requests and the outbound responses thereto,

a comparator for comparing each inbound canonical request canonical request with previously stored canonical requests in said memory to identify a matching one of said stored canonical requests that is logically equivalent to said request payload,

transmission means coupled to said comparator for returning a stored copy of that previously transmitted response in said memory that was previously transmitted in response to said matching one of said stored canonical requests.

18. The apparatus set forth in claim 17 wherein said comparator comprises:

means for generating an access key value based on the content of said inbound canonical request message;

means for retrieving selected ones of said previously received and stored canonical request messages from locations in said memory which are specified by said access key value, and

means for comparing said incoming canonical request message with said selected ones of said previously received and stored canonical requests to identify a matching one of said requests.